jamk.fi

Security monitoring in AWS public cloud

Ilkka Routavaara

Bachelor's Thesis
May 2020
Technology
Information and Communication Technology

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Description

Author(s)	Type of publication	Date
Routavaara, Ilkka	Bachelor's thesis	May 2020
		Language of publication: English
	Number of pages 46	Permission for web publication: x

Title of publication
Security monitoring in AWS public cloud

Degree programme
Information and communication technology

Supervisor(s)
Salmikangas, Esa; Hakkarainen, Pasi

Assigned by
Nixu Corporation Oyj

Abstract

Amazon Web Services has become a household name for all things *cloud*. Cloud-based solutions have seen a monumental increase in the 21st century. While cloud-based solutions have their unquestionable benefits, they come bundled with their own unique security-issues. AWS has developed several security monitoring services to combat these concerns.

A testing environment in AWS cloud was provided by a Finnish cybersecurity company Nixu Oyj. The scope was determined to involve security monitoring in AWS *public* cloud excluding private cloud and its components. The selected security monitoring tools included GuardDuty, Inspector, Security Hub and Trusted Advisor.

GuardDuty serves as a cloud-based SIEM which gathers log-data on various sources and raises findings based upon certain criteria. It oversees various areas in AWS cloud, including but not limited to account and user behavior, EC2-instance behavior, and flow statistics. Inspector is used to assess EC2 security while Security Hub collects security findings into one centralized location from different security services. Trusted Advisor provides users guidance on five different categories, one of them being security.

The goal was to research the monitoring capabilities of preselected security services and provide documentation of them and how they could be used. A set of different tests was executed in a lab-environment and the results of the tests were analyzed.

The chosen setup of security monitoring services provided a good, basic coverage on a wide variety of security subjects. It does not provide 100% coverage on everything in every environment, nor it was ever intended to do so. The setup proved to be an easy to implement easy to use-solution which did not require expert knowledge to draw great benefits from.

Keywords/tags (subjects)
AWS, GuardDuty, Inspector, Security Hub, security monitoring, Trusted Advisor

Miscellaneous (Confidential information)

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Routavaara, Ilkka	Opinnäytetyö, AMK	Toukokuu 2020
	Sivumäärä 46	Julkaisun kieli English
		Verkkojulkaisulupa myönnetty: x

Työn nimi
Security monitoring in AWS public cloud

Tutkinto-ohjelma
Tieto- ja viestintätekniikka

Työn ohjaaja(t)
Salmikangas, Esa; Hakkarainen, Pasi

Toimeksiantaja(t)
Nixu Corporation Oyj

Tiivistelmä

Amazon Web Servicestä on tullut kaikkien tuntema nimi pilvipalveluihin liittyen. Pilvipohjaiset palvelut ovat nähneet räjähdysmäisen kasvun 2000-luvulla. Vaikka pilvipalvelut tarjoavat omat kiistattomat edut, niihin liittyvät myös omat erityiset tietoturvaongelmat. AWS onkin kehittänyt useita tietoturvamonitorointipalveluita pystyäkseen vastaamaan näihin haasteisiin.

Suomalainen kyberturvallisuusyhtiö Nixu Oyj tarjosi testiympäristön AWS-pilveen. Tutkimusalueeksi määritettiin tietoturvamonitorointi AWS:n julkisessa pilvessä, poissulkien yksityisen pilven ja sen komponentit. Valitut tietoturvamonitorointi-työkalut sisälsivät GuardDutyn, Inspectorin, Security Hubin ja Trusted Advisorin.

GuardDuty on pilvipohjainen SIEM, joka kerää lokidataa eri lähteistä ja nostaa löydöksiä perustuen tiettyihin kriteeristöihin. Se valvoo eri osa-alueita AWS:n pilvessä, esimerkiksi tili- ja käyttäjäkäyttäytymistä, EC2-instanssien käyttäytymistä sekä flow-statistiikkaa. Inspectorilla arvioidaan EC2-instanssien tietoturvaa, ja Security Hub kerää tietoturvalöydöksiä eri tietoturvapalveluista yhteen keskitettyyn paikkaan. Trusted Advisor tarjoaa käyttäjille ohjausta viidessä eri kategoriassa, joista yksi on tietoturva.

Tavoitteena oli tutkia ennalta valittujen tietoturvapalveluiden monitorointikyvykkyyttä ja tuottaa dokumentaatiota niistä sekä niiden käyttämisestä. Testiympäristössä suoritettiin eri testejä ja niiden tuottamia tuloksia analysoitiin.

Valittu kokoonpano monitorointityökaluja tarjosi hyvän peruskattavuuden moniin eri tietoturvaosa-alueisiin. Se ei tarjoa 100 %:n kattavuutta kaikkiin osa-alueisiin kaikissa ympäristöissä, eikä sen ollut tarkoituskaan. Valittu kokoonpano osoittautui helposti toteutettavaksi ja helppokäyttöiseksi ratkaisuksi, josta hyötyminen ei vaatinut erityisosaamista.

Avainsanat (asiasanat)
AWS, GuardDuty, Inspector, Security Hub, security monitoring, Trusted Advisor

Muut tiedot (Salassa pidettävät liitteet)

Foreword

I would like to thank Nixu Corporation for providing me a job position and the subject for this thesis. Especially, I would like to thank Saila Suvanto, Teemu Kääriäinen and Teemu Vahtera for providing me access to the lab environment used in this thesis and, providing me with ideas and support when I needed it. Thank you.

Contents

Figures

Tables

1 Introduction

Security monitoring involves systemic information gathering from hosts, servers and networks and the analysis of that information to detect anomalies. When an anomaly is detected through either manual or automatic analysis, a response is expected. This response is usually an alarm of some kind which warrants an action. Actions can have many forms; however, they usually involve some sort of *mitigation* of a detected anomaly.

The 21st century has seen a monumental increase in cloud-based solutions and more and more companies have adopted cloud-based strategies and implementations of organizing business. Companies such as Microsoft, Google and Amazon Web Services have all invested heavily in the cloud and have developed their own cloud-based services. Currently, AWS is the biggest player in the field, but the head start it accumulated at the start of the century has dwindled as other companies have too managed to grow their market shares in recent years. (Stalcup 2020) Cybersecurity has also been another hot topic of the era, and combining these two topics together presents a unique problem landscape. When underlying infrastructure and services are bought from a cloud company, companies do not have a 100% accessibility into their environment anymore. How does one monitor something which one does not have complete access to?

To meet this dilemma, cloud-providers have developed security controls and services. AWS got a head start in developing cloud-based services against the rest of the field and thus, it is also apparent in their security services. It has developed multiple security services and technologies to offer security monitoring capabilities to their clients. At the heart of this thesis will be four of these services, namely GuardDuty, Inspector, Security Hub and Trusted Advisor. What are they? How are they used? What areas do they cover? To start, an introduction to the company that developed and provides them, Amazon Web Services.

2 Amazon Web Services – AWS

2.1 AWS – a brief history

The origins of Amazon Web Services date all the way back to the year 2002, when Amazon launched *Amazon.com Web Service* as a free service with the purpose of making it easy for developers to include Amazon.com features into their websites. Thus, began the journey of a multi-billion cloud-computing goliath AWS has become today. (Rojas 2018)

In 2003 a document describing Amazon internal infrastructure and a way to sell it as a service was released by Chris Pinkham and Benjamin Black. (History of AWS n.d.) This led to the release of *Simple Queue Service*, SQS, in 2004. Simple Queue Service is a message queuing service which provides a programmatic way for web applications to send messages. There are two implementations of SQS to choose from. The first is the standard queue which emphasizes throughput and at-least-once-delivery. The second is FIFO queue – or *first in first out* – queue which guarantees all messages are processed in the order they arrived into the service and that they are processed only once. Even though SQS is one of the oldest technologies AWS released, it is still to this day widely used by many companies globally.

These events served as a prelude to the official launch of AWS on 19th of March 2006. While SQS was only a messaging service, now it was possible for customers to use Amazon's infrastructure as the markets saw first Amazon cloud products in the form of *Simple Storage Service* – or S3 - and *Elastic Compute Cloud*, referred to as EC2. Elastic compute cloud provides server rental and hosting service for customers and it is Amazon's one of the biggest flagships to this day. (Carey 2019) In November 2008 AWS introduced *CloudFront,* Amazon's own content delivery network or CDN. This was followed by an expansion to European markets and in 2010 AWS had partnerships with the likes of Netflix and Dropbox. Amazon's cloud product family has grown substantially since then encompassing a wide variety of different fields from basic computing to application services to artificial intelligence implementations.

These services are represented in chapter 2.3 but before diving into them, first it is best to understand what *cloud* or cloud computing is.

2.2 Cloud – a meteorological phenomenon?

While everyone knows clouds that are up in the sky and occasionally block annoyingly the sun, when it comes to cloud computing it is not all that clear what it encompasses or what it is exactly. Part of that reason is that cloud computing has become such a huge, integral part of the IT industry today that it is difficult to pinpoint *exactly* "that is cloud, that is not" and set clear boundaries around it. However, there are some characteristics that do well to describe it.

Cloud computing could be described as accessing and utilizing resources and data across the Internet. When one is sitting on one's home computer and using data that is stored in computer's hard drive, that is considered *local storage*. In cloud computing the data can be stored anywhere in the world in a data center which one accesses via the Internet. Cloud computing is not about having direct physical access to hardware which stores data or provides the necessary resources, rather accessibility is guaranteed from anywhere anytime via the Internet. (Griffith 2016) Cloud providers such as Amazon provide these data centers and resources and they charge for their services based on usage.

One way of clarifying the term cloud is to subcategorize it into smaller categories. (ESDS 2018)

Cloud types

- Public cloud: Infrastructure provided by a cloud computing company that offers it as a service to other organizations. Available to any organization or company.
- Private cloud: Infrastructure hosted by a party itself, only accessible to said party. Also known as enterprise cloud.
- Hybrid cloud: Utilizes both public and private cloud. A company can for example host external resources in public cloud and internal resources in private cloud.
- Community cloud: Cloud shared between different organizations with a common goal or community.

Additionally, regardless of the cloud type there is a wide variety in cloud computing services. These cloud computing services are listed below.

- IaaS: Infrastructure-as-a-service. Cloud provider rents its infrastructure to customer. This can be servers, virtual machines or other hardware.
- PaaS: Platform-as-a-service. Designed for customers who want to develop, run or manage their applications without worrying about the underlying necessary infrastructure, such as databases and storage.
- SaaS: Software-as-a-service. Software delivered over the Internet. Can be either on-demand or subscription based.
- FaaS: Functions-as-a-service. Developers can upload blocks of code that only execute when certain criteria are met.
- BaaS: Backup-as-a-service: Customers can purchase backup and recovery services for their data.

Because of its versatility and scalability, cloud computing has revolutionized the IT-industry. While private citizens might simply enjoy the advantages cloud computing has brought to the IT industry in the form of backing up their personal data or enjoying Netflix, these advantages are particularly apparent in corporate environments. First off, hosting services in public cloud eliminate the need to acquire and maintain one's own hardware, which cuts down costs considerably. Service scalability is improved since the right amount of resources is always utilized based on demand which is also reflected in enhanced performance; if there are peaks in demand more resources are deployed automatically, and users do not experience any downtime due to overworked servers. Private clouds offer companies strict security controls but that is just not limited to private clouds; public clouds offer a wide variety of different technologies and policies to protect customers from potential security threats. Since cloud providers handle cumbersome IT management chores, companies can only focus on their business goals, which translates into boosted productivity. Cloud solutions also make it easy to backup customer and corporate data. (Cloud computing benefits 2020) This can be a game saver for companies that might be unfortunate victims of a e.g. a ransomware attack, which have been prevalent across multiple sectors the past years. (Davis 2019)

2.3 AWS technologies and services

Starting from SQS, Amazon has since progressed into various fields comprising a whole army of different technologies and services. In the following are listed just some of the fields with technologies and services associated with them. A full service catalog can be found in https://aws.amazon.com/servicecatalog/?aws-service-catalog.sort-by=item.additionalFields.createdDate&aws-service-catalog.sort-order=desc, this, however, requires an AWS account with a valid credit card associated with the account.

In computing AWS offers one of its flagships that was touched upon at the beginning of this chapter, EC2 or Elastic Computing Cloud. It is a virtual machine where the user has OS level control, and anything can be run in it. For beginners AWS has developed *LightSail*. LightSail focuses on automating the required underlying components, e.g. networking, storage so that the user can solely focus on the application that is served from it. Worth mentioning are also *Elastic Container Service* and *Elastic Container Service for Kubernetes*. ECS is a scalable container service and with EKS user can run Kubernetes on AWS without installing Kubernetes control plane. With *Lambda* users can run functions in the cloud.

For storing AWS offers the aforementioned *Simple Storage Service* or S3 and *Elastic File System.* EFS provides file storage for EC2 instances. For database users can choose *Relational Database Service* which allows to run relational databases (such as MySQL or MariaDB) or *DynamoDB* which in turn is a NoSQL database. It is designed to be a high-performance database with very small latency. *Elasticache* was developed to cache queries to offer a form of load balancing to database servers. In case there is a need to migrate a database to AWS, *DMS* or *Database Migration Service* is reliable choice. Servers can be migrated using *Server Migration Service* (SMS).

Content delivery networks have become very popular in the 21st century and Amazon offers *CloudFront* as their solution. Edge locations are CDN endpoints that cache resources that are then served at a faster pace to users. Regarding networking worth mentioning are *Virtual Private Cloud* and Amazon's DNS-service *Route53.* Developers

have a wide variety of services to choose from. *CodeStar* was created as a cloud-based service for software development. Developers can use *Cloud9*, a cloud-based IDE for development also. Furthermore, Amazon offers *CodeCommit* as a version control service. To manage AWS projects and instances there are *CloudWatch* and *Config* notably. CloudWatch is a tool which monitors AWS instances and can trigger alarms based on different conditions. Config alarms users when there are issues in their configurations. *Identity and Access Management* is used to manage users and group and assign policies related to them. Lastly, *Simple Notification Service* can be used to send bulk notifications both in email and SMS format. (Yadav 2018)

These were just some of the technologies and services to choose from. Amazon has also expanded to many other fields, including mobile services, streaming services, artificial intelligence, virtual reality, game development and IoT to name just a few. Going through the whole Amazon service catalog would be a thesis subject of its own and it is not really in the scope of this thesis. It suffices to say that whatever use case one can come up with regarding IT, there is a good chance AWS has some service or tool designed for that purpose. The actual scope of this thesis is the security aspect of AWS public cloud, which is why in this chapter AWS' security-related technologies and services were not represented. These are the subject of the next chapter.

3 AWS Security

3.1 AWS Security components

As could be expected, just as AWS had a wide variety of different services encompassing multiple fields, the same goes for security. The scope of this thesis is public AWS cloud and it is in this context that services which got a more detailed inspection were chosen. These services are *Inspector*, *GuardDuty, Security Hub* and *Trusted Advisor.* They are covered in upcoming subchapters; however, before delving into them, first a quick general overlook of AWS security follows.

In a 2019 AWS re:Inforce talk Becky Weiss introduced a summary of AWS security. This summary is depicted in Figure 1. (Weiss 2019) As can be seen, AWS security can be divided into three categories: permission management, data encryption and networking.

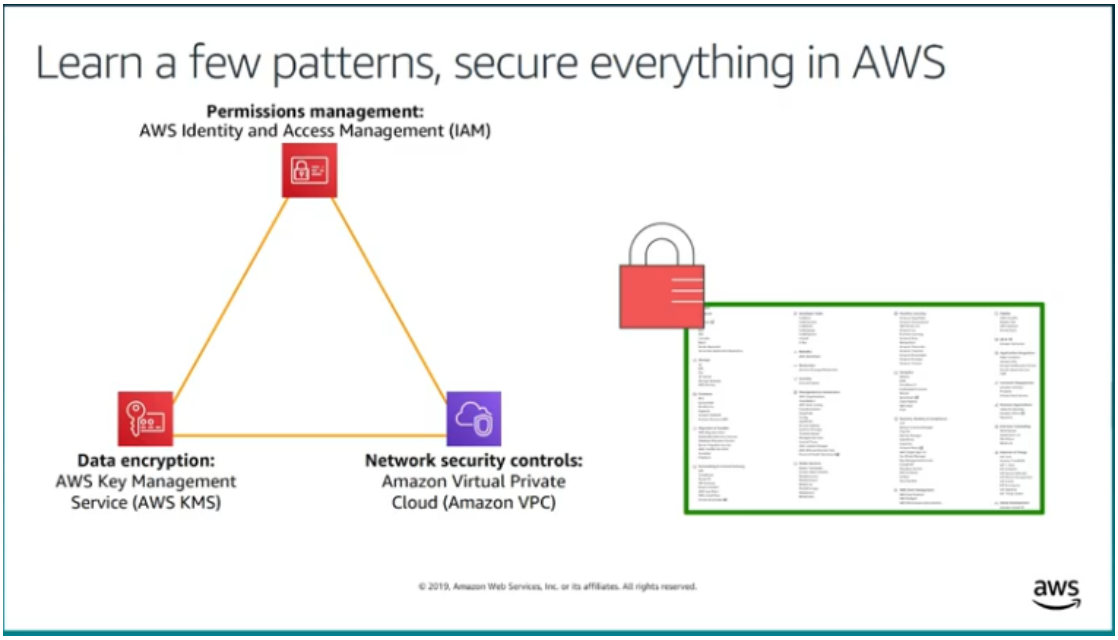


Figure 1. AWS Security components

Identity and Access Management has a twofold function. Firstly, it authenticates users, i.e. it verifies users' identities by asking them their credentials when signing in. Secondly, it authorizes users, which means assigning permissions to users. Authorization takes place after authentication. With AWS IAM users can e.g. share access to their AWS accounts, grant permissions to users and groups and provide credentials for applications and require two-factor authentication from users. (What is IAM 2020)

AWS Key Management Service is responsible for creation and management of cryptographic keys. These keys are used to encrypt and decrypt data in AWS. KMS can be managed either through management console or AWS KMS API (application programming interface). For traceability it is integrated with *CloudTrail* which logs key usage. (AWS KMS 2020) Lastly, networking and particularly AWS' Virtual Private Cloud enables user to secure their networks. VPC serves as network layer (OSI model layer 3) for EC2 instances. Key concepts of VPC include subnet, route table, internet gateway and VPC endpoint. VPC endpoint makes it possible for VPCs to connect to AWS services without the need of a gateway or similar device in between. (AWS VPC 2020) These concepts are demonstrated in Figure 2 (VPC 2020).

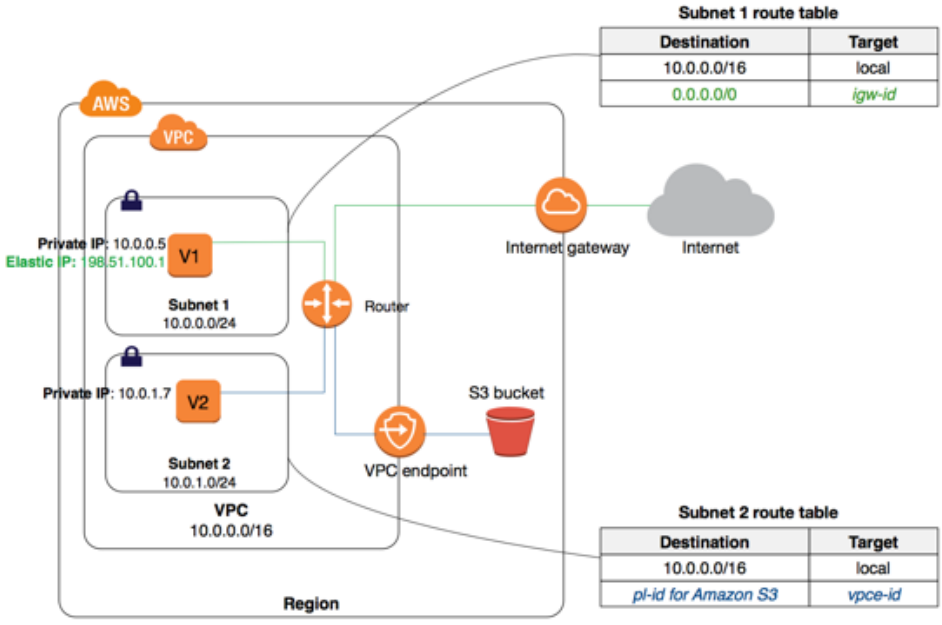


Figure 2. AWS VPC Logical topology example

3.2 AWS Security services

AWS WAF

AWS Web Application Firewall's function is to monitor, and filter HTTP/HTTPS requests made to Amazon API Gateway, CloudFront or Application Load Balancer. It gives users flexibility into content access management. Users can set all requests to be blocked except for the ones they specifically sanctioned or all requests to be allowed except ones that were explicitly denied. Deciding between the two depends on the use case. It serves as protection against attacks originating from the web. This is achieved through predefined conditions. These conditions can be based on a vast range of characteristics e.g. IP ranges, geolocational information, payload content and length, header values or regex patterns. These rules can also be counter-based, meaning that depending on how many times certain request is made in a time frame whether it gets blocked or not. (AWS WAF 2020)

AWS Shield

Shield is DDoS (Distributed Denial of Service) protection tool. It has the standard version which comes for free for all AWS customers and advanced version. The advanced version offers enhanced DDoS protection for EC2 instances and other AWS services. While the standard version only protects network and transport layer (layers 3 and 4) attacks the advanced version offers protection also against layer 7 attacks (application layer). A major benefit of the advanced version is also that it deploys network ACLs in midst of an attack to network borders. Network ACLs commonly offer protection against traffic volumes limited to interface processing power but when deployed to AWS border, they can protect against much bigger volumes of traffic. Users who opt for the advanced version also get AWS WAF for free. (AWS Shield 2020)

AWS Macie

Macie is a security service designed to protect sensitive information in S3 instances. Currently S3 is only supported but Amazon is developing support for other data stores as well. It uses machine learning to classify and protect sensitive information,

such as personal information. Once Macie discovers anomalous activity related to information it considers sensitive, an alert is raised. These alerts can then be processed in Macie's dashboard or they can be sent to CloudWatch for further analysis and custom responses. Macie provides enhanced management capabilities for users regarding their data. (Macie 2020)

Amazon CloudWatch

CloudWatch monitors users' applications, infrastructure, and resources. It collects data regarding them in various forms: log-data, metrics (measurable variables) and events. User can customize their own metrics tailored to meet their specific needs. This can be e.g. available disk space of an EC2 instance. Additionally, users can set alarms to be triggered once certain criteria are met. This can be e.g. when a metric exceeds a predefined threshold. Once an alarm goes off, users can plan automatic responses to remediate these alarms. Continuing the previous example when CloudWatch alarms that disk space is running out on an EC2 instance, another instance can be set to be automatically deployed in conjunction with the first instance. Users can utilize CloudWatch in various ways. It has its own console, command line interface and API. With CloudWatch users have a complete visibility into their system; in what quantity and how resources are being used, how applications are performing and what is the overall operational status of the system. (AWS CloudWatch 2020) Figure 3 showcases an example implementation. (CloudWatch 2020)

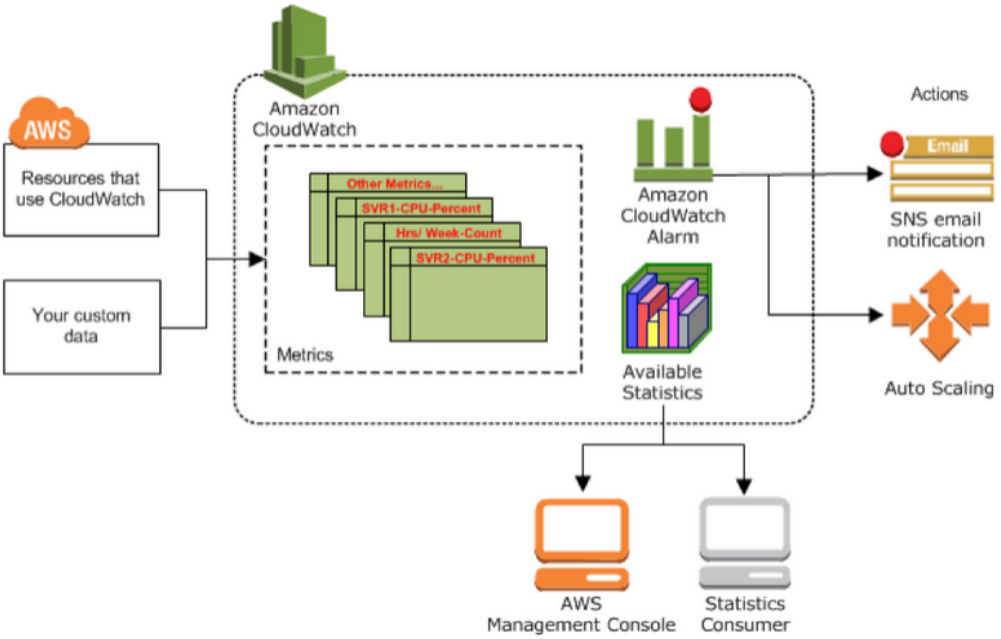


Figure 3. CloudWatch example use case

3.3 Inspector

Amazon Inspector is a tool developed to conduct security assessments on Amazon EC2 instances. Using Inspector, it is possible to deploy large scale security assessments where the exact scope is set by user himself or inspect singular instances. To achieve this, Inspector agents are installed on hosts. The agent monitors system behavior such as processes and collects telemetry data, e.g. configuration data. Each assessment provides user a report where potential security risks and misconfigurations are listed. These assessments can be run manually, or they can be automated to run on certain intervals. Users can also automatize the remediation process of found security issues by Inspector. In a blogpost Eric Fitzgerald offers one implementation of this which starts by Inspector agent being installed on an instance via EC2 Systems Manager. Next, an SNS topic is set up where assessment findings are sent. Then, a Lambda-function is configured which firstly fetches the information from the topic and secondly based on the information received, remediates found issues through Systems Manager. (Fitzgerald 2018)

Inspector has a built-in library which consists of rulesets and reports. These include industry best practices and compliance standards alongside vulnerability intelligence. These are the building blocks which security assessments are run against. Amazon has a dedicated team whose sole responsibility is to keep these rulesets and reports up to date. Each security issue or misconfiguration listed in post-assessment report is coupled with recommendations how to mitigate or fix them. This is one of the strongest features of Inspector; users who even might not be security-savvy can keep their EC2 instances secure and in line with industry best principles and practices. It is a powerful tool for the uninitiated and initiated alike. While more experienced users might use it for system activity monitoring and security assessment automation, beginners get valuable assistance in configuring their instances properly.

The configuration of a security assessment is done through *assessment template*. In the template key properties are set, such as which rules are used, where potential findings are sent and how long the assessment can take. Before setting up the template IAM-role needs to be set up, which allows Inspector to access instances. Next, the scope of the assessment is determined, i.e. which instances are to be assessed. Instances that are in the scope are *tagged* and agents are deployed to them. Inspector has its own browser-based console where users can implement their own assessments. To automate the process of running security assessments users can write their own scripts using AWS command line tools. Inspector has its own SDKs and API for users who wish to interact with it programmatically. (Inspector 2020)

3.4 GuardDuty

Detecting existing and emerging threats has been a part of cybersecurity landscape for decades and judging current trends, its relevance certainly does not seem to be declining for foreseeable future. GuardDuty is Amazon's answer to this reality. SIEM (Security Information and Event Management) products have seen a quick rise in cybersecurity field in the past decade. At the core, they are designed to take in huge masses of log data from multiple endpoints, analyze that data and produce predefined outputs such as alerts. GuardDuty shares these characteristics. It utilizes many

of the high-level developments in the current IT climate: machine learning, anomaly detection and threat intelligence.

Collecting log data is at the core of GuardDuty's functionality. All detections are based on certain predefined criteria and those criteria are based upon log data. GuardDuty has three sources to collect this log data. They are CloudTrail event logs, VPC flow logs and DNS logs. CloudTrail tracks AWS API call history for a user account. It takes note of the entity (AWS service or user) that performed the call, source IP and timestamp. CloudTrail can also be set to collect management events, such as assigning permissions for an IAM role for certain resource. These are then forwarded to GuardDuty for further analysis. VPC flow logs collect information about ingress and egress traffic on VPC interfaces (e.g. IP addresses, protocol information). Finally, DNS logs deal with DNS requests made and DNS responses received, i.e. mapping of IP-addresses and domain names. Depending on the log data received, GuardDuty generates *findings* (security issues) when received logs fulfill certain criteria or exceed certain thresholds such as POST requests made from the same source within a time frame.

GuardDuty can be enabled on each AWS account. Billing is based upon log volume that GuardDuty analyses. Users can invite other accounts to become associated with their account in GuardDuty. That account becomes then master GuardDuty account and other accounts are members. Master can then use GuardDuty to monitor other users as well. One master account can only be associated with thousand members which is one of GuardDuty's shortcomings. (GuardDuty 2020) Another one is that users cannot make their own custom findings but are limited to what GuardDuty has built in. Of course, it is no small pool but different environments with different configurations sometimes need highly specified solutions, which at least currently is not possible to do.

3.5 Security Hub

Amazon has a vast range of security products for their users to choose from. Managing them in users' environments can get quite quickly quite complicated. Also, getting them to work together in a synchronized manner is no trivial matter either. Security Hub is for this purpose. It can be thought of as a centralized management tool for different security products in AWS environment. As an example, both Inspector and GuardDuty can be integrated to it. It has support for third party vendors as well. Security Hub takes in security alerts and issues en masse and organizes them based upon customizable configurations. It also performs automated compliance checks in the background and raises findings if something is not in line with best practices.

One of the advantages of using Security Hub is that it uses a standardized format for security issues it receives from different security services called *AWS Security Finding Format*. Another is that users can get a holistic view of their security environment and spot trends even when the security findings come from multiple different sources. However, perhaps the biggest advantage is that Security Hub can be integrated with CloudWatch, which enables the automatization of mitigation processes of found security issues and concerns. There is a high degree in customizability of actions that can be taken automatically when a security finding emerges. Users can send findings e.g. automatically to SIEMs or SOAR (Security Orchestration Automation and Response), ticketing systems or SNS. Just as GuardDuty, Security Hub can be enabled in any account and users can invite other users to become associated with their Security Hub-service. The member limit is also one thousand.

Security Hub collects security findings across AWS environments from different security services. These findings are then organized in Security Hub and different actions can be taken for each finding. After a finding has been created it has a 90-day retention time if it does not get updated. Users however have the option of archiving findings which do not get deleted. Users can also use *insights* which are a group of related security findings to form a better general view of what needs their attention. Security Hub offers its own inherent findings, but users can produce their own customized findings also. (Security Hub 2020)

3.6 Trusted Advisor

While Trusted Advisor is labeled as a management and governance service by Amazon, one of its functionalities is based around security. Launched in 2013, Trusted Advisor was designed to offer guidance and industry best practices across different fields in AWS. Today, it is comprised of five categories: cost optimization, performance, fault tolerance, service limits and security. Trusted Advisor is a part of the AWS management console and all users currently get seven of core checks free of charge. (Trusted Advisor 2020) Users can also turn on Trusted Advisor Notifications free of charge to receive emails regarding Trusted Advisor findings. To get all the functionality and checks, users have the option to choose business and enterprise support. Trusted Advisor is part of the AWS Support brand which means users can interact with it through an API also instead of the management console.

The free version comes with several security checks: S3 bucket permissions, IAM use and MFA on root account to name a few. Paid versions expand on these areas and introduce a wide variety of other security-related areas such as key management and SSL certificates. (Best practice checklist 2020) To get the most out of Trusted Advisor checks, Amazon has enabled check results to be sent directly to CloudWatch. There, custom rules can be made specific to these results, and a wide variety of actions can be triggered. For example, users can automate mitigation processes or send notifications. Amazon has coined a thirteen-step guide for their users where a CloudWatch events rule is created for Trusted Advisor. The whole process can be found on Amazon website. (CloudWatch events 2020)

Security considerations aside Trusted Advisor can be a fantastic tool in cost optimization. For non-experienced users this can save a pretty penny. On top of cost optimization, more experienced users can take valuable intake from Trusted Advisor regarding system performance and stability. Combining all the technologies described in this chapter, it is no wonder why so many major corporations, institutions and financial sector entities have chosen Amazon as their go-to partner. It provides a very secure platform with many inherent security features while providing a great deal of flexibility and customizability to security implementations.

4 Practical glances

Before delving into how these tools can be used, the actual environment in which these tools are showcased in this thesis should be mentioned. The environment is a testing platform owned by Nixu Oyj. At the time of writing this thesis, it was a very new project used for integration testing and development. Hence, findings that are shown moving forward should be considered only as such, caused by testing and development. In chapter 5, an EC2-instance is set up and hardened using Inspector. Chapter 6 showcases GuardDuty through one use case.

Starting with GuardDuty, in Figure 4 the opening page of GuardDuty is showcased. It has all findings listed associated with the account.

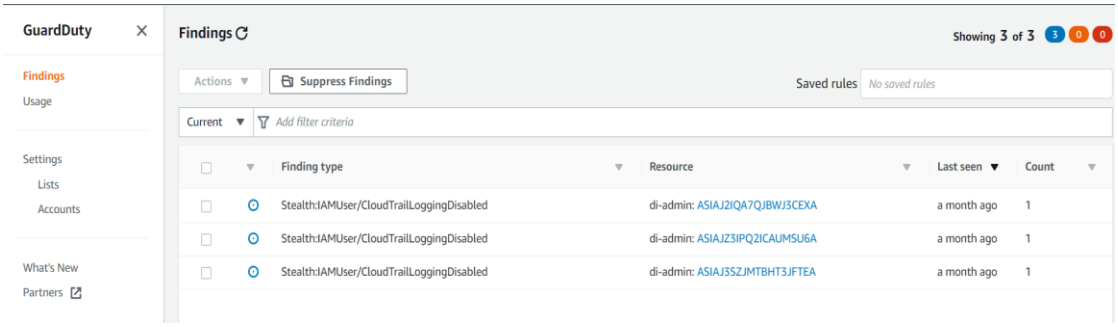


Figure 4. GuardDuty findings view

GuardDuty categorizes findings according to *types*. Table 1 showcases all finding types associated with their findings, respectively. As was discussed in chapter 3.4, unfortunately making one's own customizable findings is not supported in GuardDuty. Users are limited to what AWS provides.

Table 1. GuardDuty finding types (GuardDuty findings 2020)

FINDING TYPE	FINDINGS
Backdoor Finding Types	• Backdoor:EC2/Spambot • Backdoor:EC2/C&CActivity.B!DNS • Backdoor:EC2/DenialOfService.Tcp • Backdoor:EC2/DenialOfService.Udp • Backdoor:EC2/DenialOfService.Dns • Back-door:EC2/DenialOfService.UdpOnTcpPorts • Back-door:EC2/DenialOfService.UnusualProtocol
Behavior Finding Types	• Behavior:EC2/NetworkPortUnusual • Behavior:EC2/TrafficVolumeUnusual
CryptoCurrency Finding Types	• CryptoCurrency:EC2/BitcoinTool.B!DNS • CryptoCurrency:EC2/BitcoinTool.B
PenTest Finding Types	• PenTest:IAMUser/KaliLinux • PenTest:IAMUser/ParrotLinux • PenTest:IAMUser/PentooLinux
Persistence Finding Types	• Persistence:IAMUser/NetworkPermissions • Persistence:IAMUser/ResourcePermissions • Persistence:IAMUser/UserPermissions
Policy Finding Types	• Policy:IAMUser/S3BlockPublicAccessDisa-bled • Policy:IAMUser/RootCredentialUsage
PrivilegeEscalation Finding Types	• PrivilegeEscalation:IAMUser/Administra-tivePermissions
Recon Finding Types	• Recon:EC2/PortProbeUnprotectedPort • Recon:EC2/PortProbeEMRUnprotectedPort • Recon:IAMUser/TorIPCaller • Recon:IAMUser/MaliciousIPCaller.Custom • Recon:IAMUser/MaliciousIPCaller • Recon:EC2/Portscan • Recon:IAMUser/NetworkPermissions • Recon:IAMUser/ResourcePermissions • Recon:IAMUser/UserPermissions
ResourceConsumption Finding Types	• ResourceConsumption:IAMUser/Com-puteResources

Stealth Finding Types	• Stealth:IAMUser/S3ServerAccessLogging-Disabled • Stealth:IAMUser/PasswordPolicyChange • Stealth:IAMUser/CloudTrailLoggingDisabled • Stealth:IAMUser/LoggingConfigurationModified
Trojan Finding Types	• Trojan:EC2/BlackholeTraffic • Trojan:EC2/DropPoint • Trojan:EC2/BlackholeTraffic!DNS • Trojan:EC2/DriveBySourceTraffic!DNS • Trojan:EC2/DropPoint!DNS • Trojan:EC2/DGADomainRequest.B • Trojan:EC2/DGADomainRequest.C!DNS • Trojan:EC2/DNSDataExfiltration • Trojan:EC2/PhishingDomainRequest!DNS
Unauthorized Finding Types	• UnauthorizedAccess:EC2/MetadataDNSRebind • UnauthorizedAccess:IAMUser/TorIPCaller • UnauthorizedAccess:IAMUser/MaliciousIPCaller.Custom • UnauthorizedAccess:IAMUser/ConsoleLoginSuccess.B • UnauthorizedAccess:IAMUser/MaliciousIPCaller • UnauthorizedAccess:EC2/TorIPCaller • UnauthorizedAccess:EC2/MaliciousIPCaller.Custom • UnauthorizedAccess:EC2/SSHBruteForce • UnauthorizedAccess:EC2/RDPBruteForce • UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration • UnauthorizedAccess:IAMUser/ConsoleLogin • UnauthorizedAccess:EC2/TorClient • UnauthorizedAccess:EC2/TorRelay

Figure 4 demonstrates that three Stealth:IAMUser/CloudTrailLoggingDisabled findings concerning three different resources have been listed. Clicking on the resource, a more detailed view is presented offering more information on the event, as can be seen in Figure 5. Account ID has been blurred deliberately here and in subsequent Figures as well.

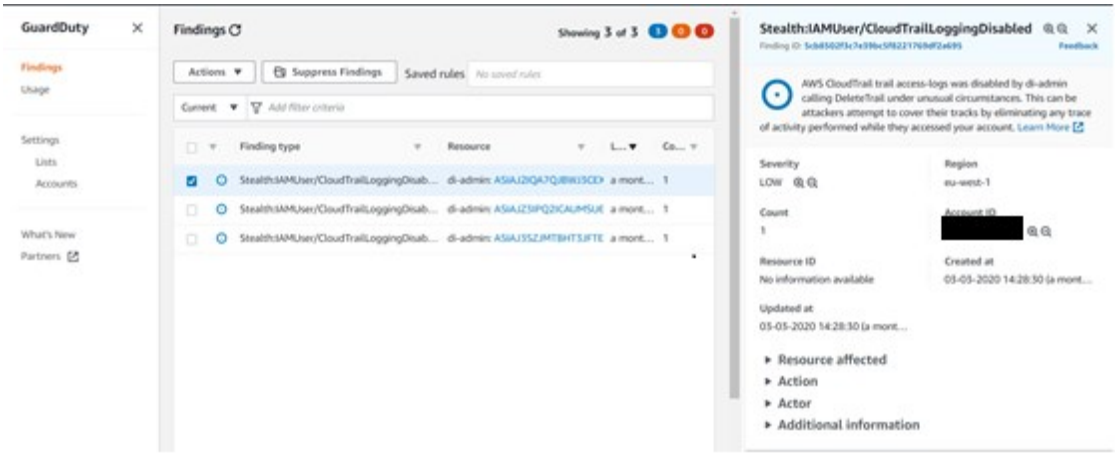


Figure 5. Example GuardDuty finding

From this view users can choose “Learn More” which forwards them to Amazon documentation of the related finding. It contains remediation steps for the finding to help users mitigate emerged threats and issues.

Security Hub collects security related issues across multiple different services and serves as a centralized management tool for services that have been integrated into it. Figure 6 shows all the main components of Security Hub.

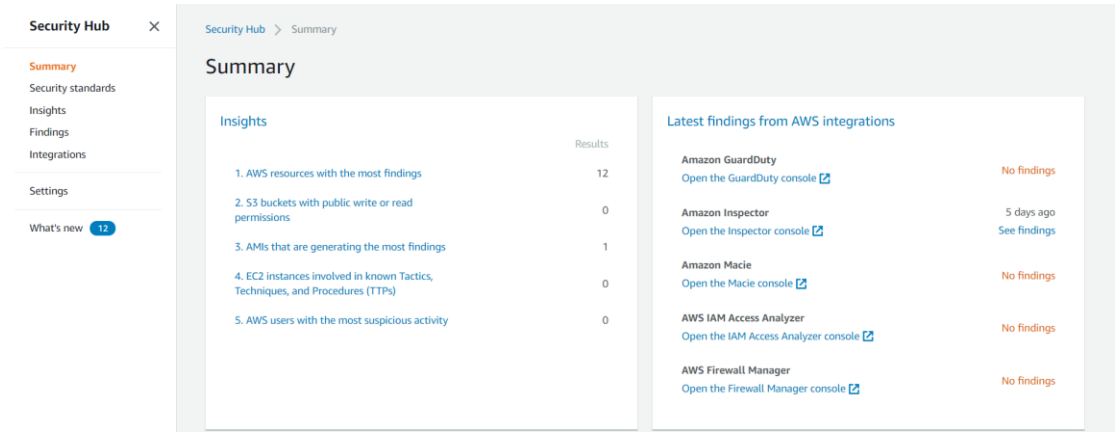


Figure 6. Security Hub Console

From integrations tab users can configure Security Hub to accept findings from AWS services and from other vendors also. Splunk, Symantec, Sophos and McAfee are just

a few of the supported vendors. Each vendor and service have an integration configuration guide attached to it to help users navigate through the integration process as painless as possible. Forwarding security findings to Security Hub can be disabled also here as can be seen in Figure 7.

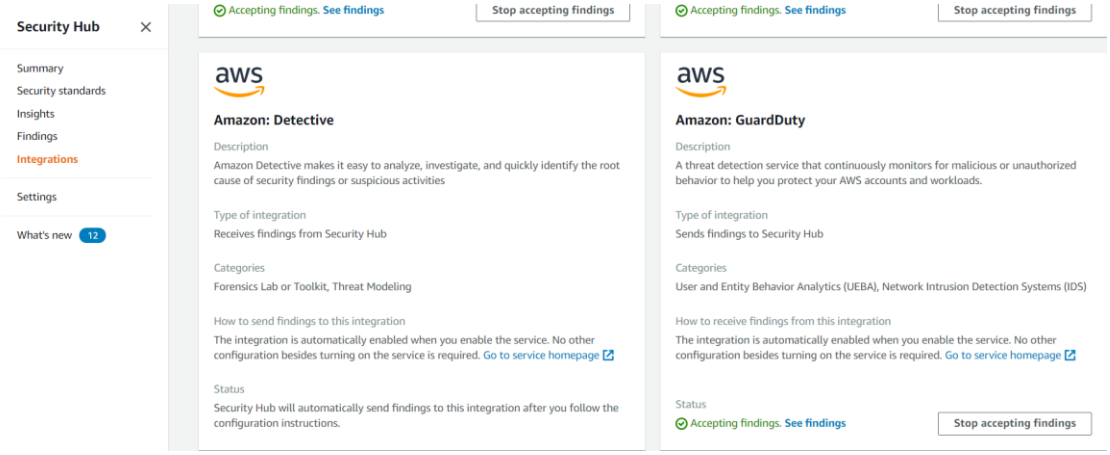


Figure 7. Security Hub Integration management

Findings-tab contains all the findings that have been forwarded to Security Hub. They can be reviewed the same way they could be reviewed in their original service from which they were sent to Security Hub by clicking the title of the finding. Figure 8 showcases one finding which was generated by Inspector alongside with its remediation steps.

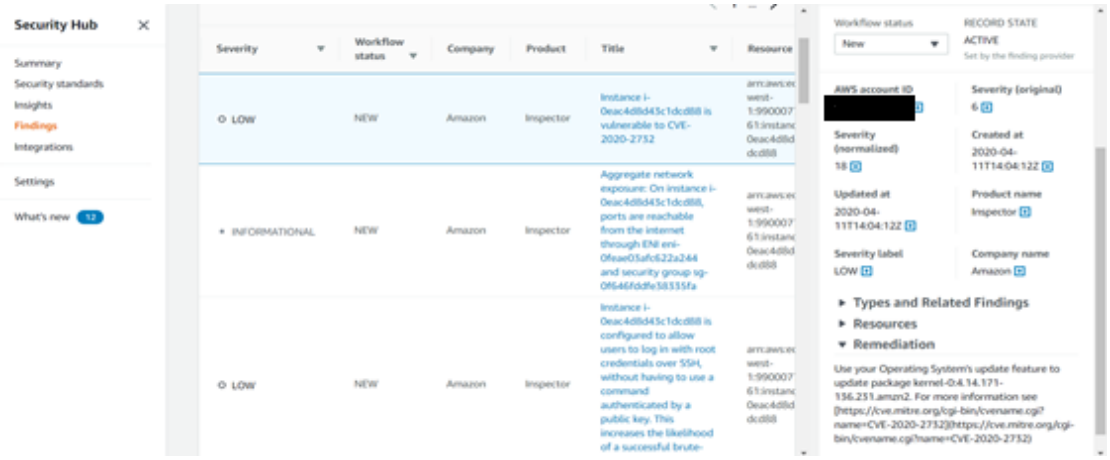


Figure 8. Inspector finding in Security Hub

Insights are filters for findings. AWS offers multiple default built-in insights called managed insights, yet, users can also customize their own. This feature can become extremely useful in cases where there are thousands of instances and multiple services running which all send findings into Security Hub, such as when a company has its infrastructure running in AWS. Some of the managed insights can be seen in Figure 9.

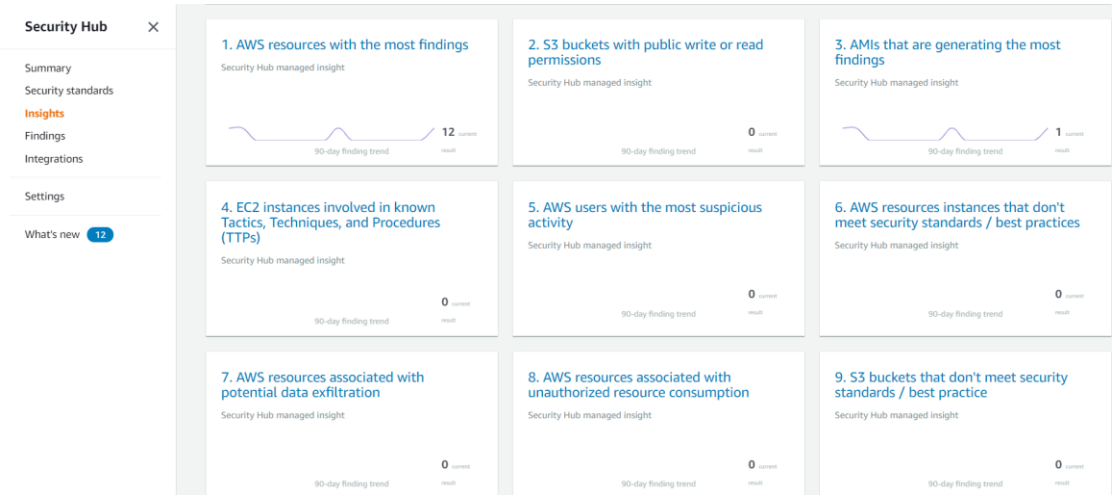


Figure 9. AWS managed insights

Perhaps one the strongest features of Security Hub is the use of automated checks against security standards. Currently, AWS offers two standards which can be enabled from "*Security standards*" -tab shown in Figure 10.

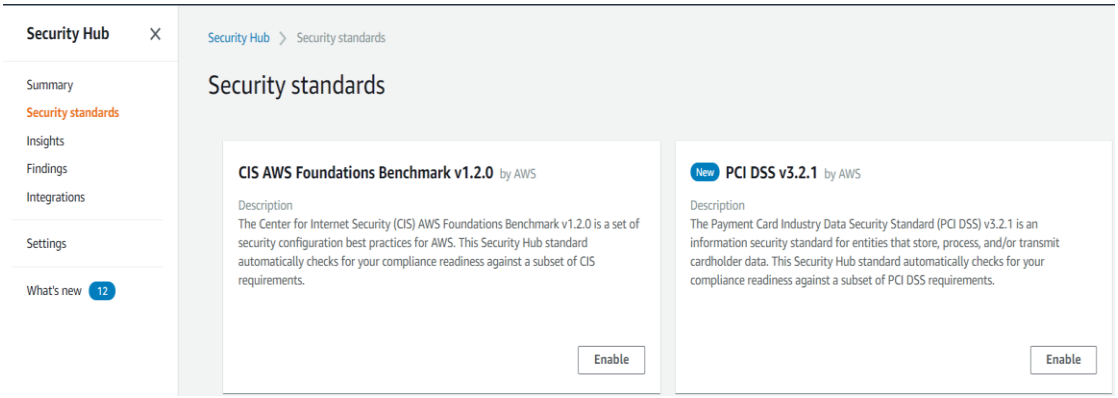


Figure 10. Security Hub Security standards

These are extremely comprehensive automated tests that are comprised of various rules and compliance checks. Once enabled they produce their own findings and give a readiness score. Additionally, they identify problematic accounts and resources that require mitigative actions to take place. (Security Hub Standards 2020) Users running large-scale deployments in AWS can find relief in that Security Hub can be configured with CloudWatch and AWS Lambda to fully automate mitigation processes.

Trusted Advisor provides additional security checks to further protect users' accounts and instances, but it does not just limit there. Figure 11 showcases the five categories introduced in chapter 3.6.

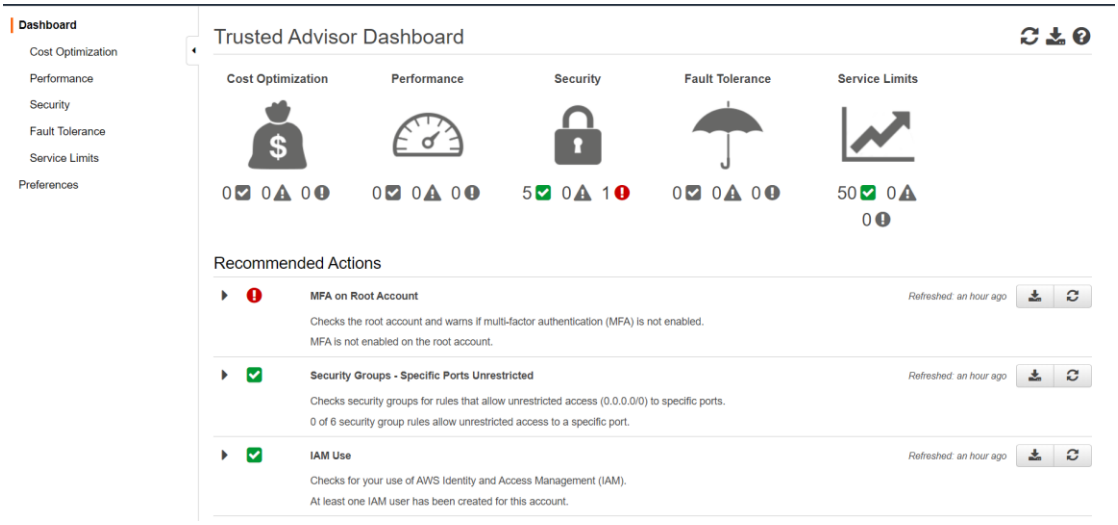


Figure 11. Trusted Advisor Console

By default, all users get Trusted Advisor's Basic support plan when registering an AWS account. However, to get access to all five categories users need to upgrade either to developer, business or enterprise support plan which are separately billable. Large-scale deployments can really take great benefits from these. For example, in the field of cost optimization it can get quite hard quite quickly to keep track of every instance and how resources are being utilized on a larger scale. Businesses starting out can also greatly cut down unnecessary, "blind" expenses at the start of their lifecycle when maintaining a healthy balance between expenses and proceeds is of

utmost importance. Since the testing environment had only the basic plan only security and service limits -categories were available. Figure 12 lists all security issues found by Trusted Advisor.

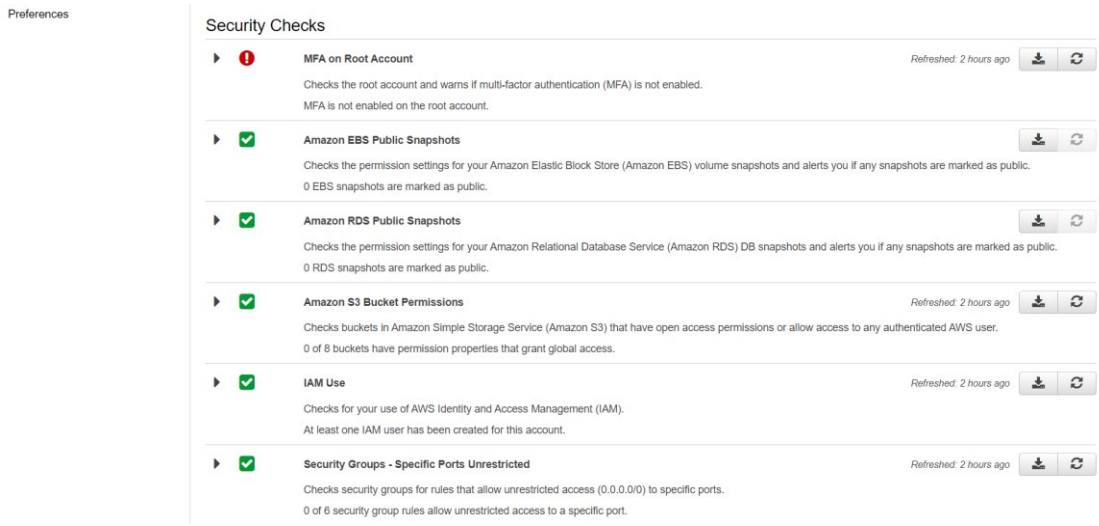


Figure 12. Security findings by Trusted Advisor

Users can further investigate found issues by expanding the wanted issue. Additional information links and most importantly, recommended actions are listed as illustrated in Figure 13.

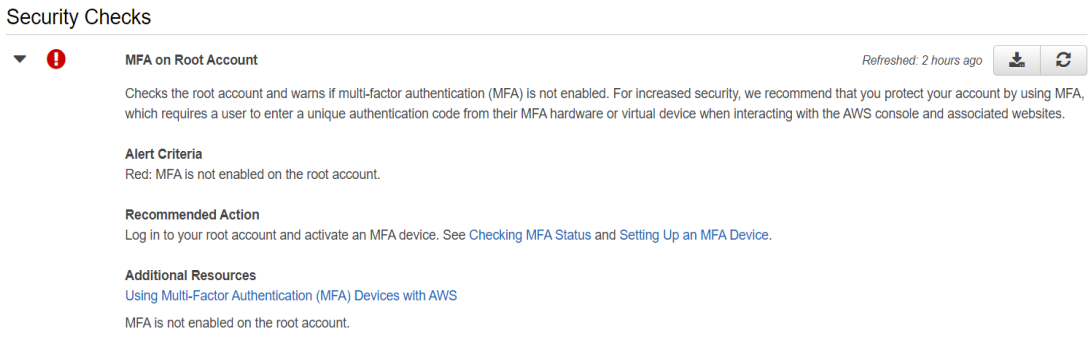


Figure 13. Viewing security findings in Trusted Advisor

Service limits-tab contains checks in similar fashion as Security-tab, and they can be expanded the same way for further investigation and recommended actions. Since this thesis focuses on security in AWS public cloud, going deeper into Service limit checks is not within the scope of this thesis.

Inspector has its own console view also. Users can set up their own assessment targets, templates and runs (execute security assessments) as depicted in Figure 14.

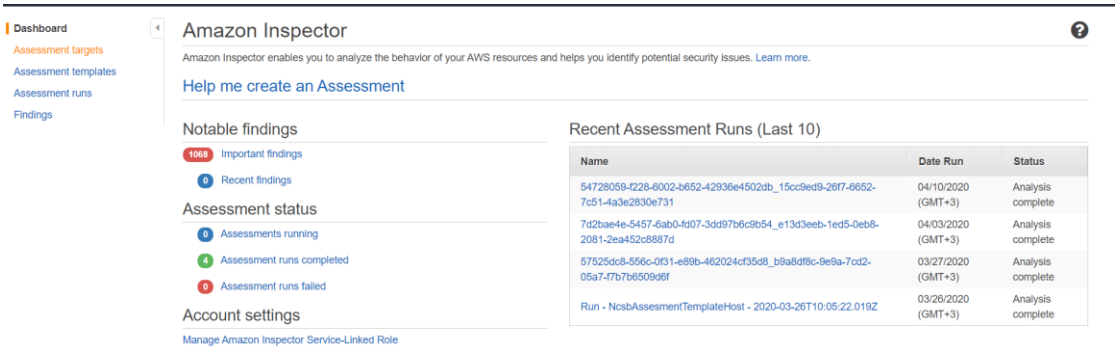


Figure 14. Inspector console

More detailed view of Inspector will be provided in the next chapter where a fresh new EC2-instance is hardened using Inspector.

5 Securing EC2-instance with Inspector

5.1 Setting up own security assessment using Inspector

To successfully run Inspector assessments on EC2-instances, inspector agents need to be installed on them first. This can be done in many ways. When defining an assessment, as will be done later, users can choose to install agents on all EC2 instances and AWS will automatically take care of installations on behalf of them. Another way is to install them through *Systems Manager Run Command.* Finally, users can install agents themselves manually, as is the case here. (Inspector agents 2020)

After connecting to EC2-instance user simply needs to run following commands as shown in Figure 15.

```
[ec2-user@ip-192-168-1-14 ~]$ wget https://inspector-agent.amazonaws.com/linux/latest/install
--2020-04-24 14:02:51--  https://inspector-agent.amazonaws.com/linux/latest/install
Resolving inspector-agent.amazonaws.com (inspector-agent.amazonaws.com)... 13.224.64.227, 2600:9000:21ca:a800:
4:6195:f549:e8e1, 2600:9000:21ca:bc00:4:6195:f549:e8e1, ...
Connecting to inspector-agent.amazonaws.com (inspector-agent.amazonaws.com)|13.224.64.227|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 30215 (30K)
Saving to: 'install'

install                     100%[=======================================>]  29.51K  --.-KB/s    in 0s

2020-04-24 14:02:51 (283 MB/s) - 'install' saved [30215/30215]

[ec2-user@ip-192-168-1-14 ~]$ sudo bash install
```

Figure 15. Installing Inspector agent on EC2-instance

Next, an assessment is defined where the scope and other key configurations are determined. This is done in AWS Inspector console. First, assessment targets (instances that will be in the security assessment) are chosen, and a name of the assessment is given. Users have the option to choose all EC2 instances that are under current AWS account and install Inspector agents on them automatically in this view. Earlier Inspector agent was installed manually, so this is not required in this example. Since only one instance is going to be inspected, a tag needs to be added to it which is how

AWS identifies unique instances. A tag consists of key-value pairs. Users can add tags to their instances in AWS EC2 console view. Figure 16 shows described steps.

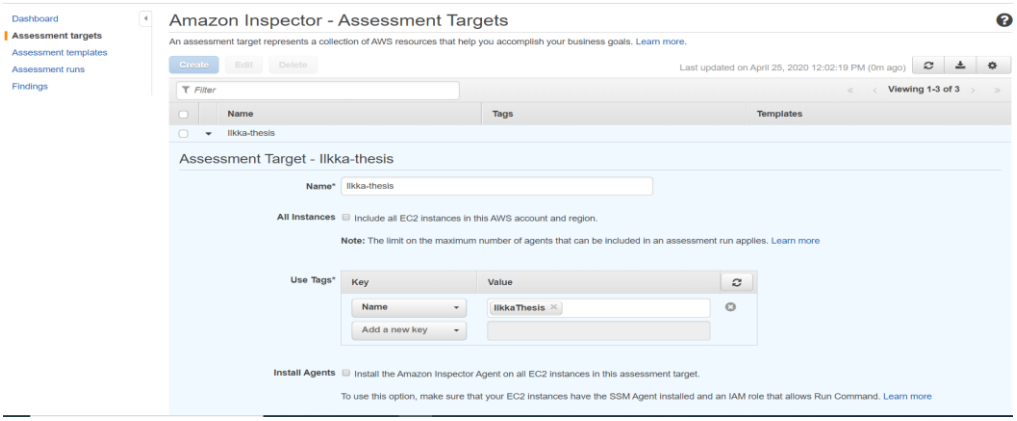


Figure 16. Setting up assessment targets

Following up is defining an assessment template. Assessment template can be thought of as a runbook which dictates what instances are assessed, the duration of the assessment and what rule packages are used in the assessment among other things. Rule packages are consisted of different industry standards, best practices and rules which ultimately dictate what security checks are done. AWS has a default recommendation of one hour. In Figure 17 these options are highlighted. It should be noted that in *Target name*-field the name of our previously generated assessment target is given.

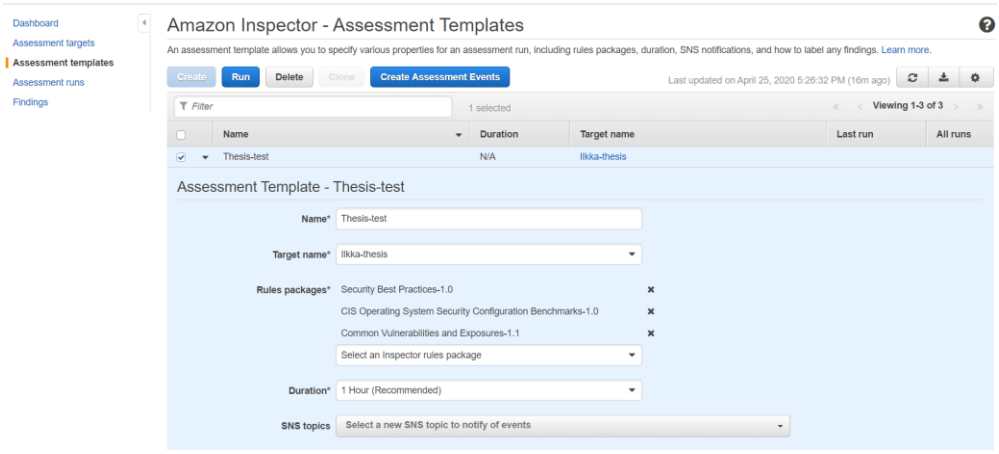


Figure 17. Defining an Inspector assessment template

SNS topics is used in case where the results of an assessment are chosen to be forwarded to other service or services. This can be used in e.g. automating the mitigation process of found security issues as was briefly discussed in chapter 3.3. This will not be done in this example, however. Additional configurations include adding a tag to the template, adding attributes to findings, and making the assessment run on certain intervals as shown in Figure 18. These options will not be used in this example. Once all obligatory and voluntary options are set, *Create and run* will save the template and start the assessment.

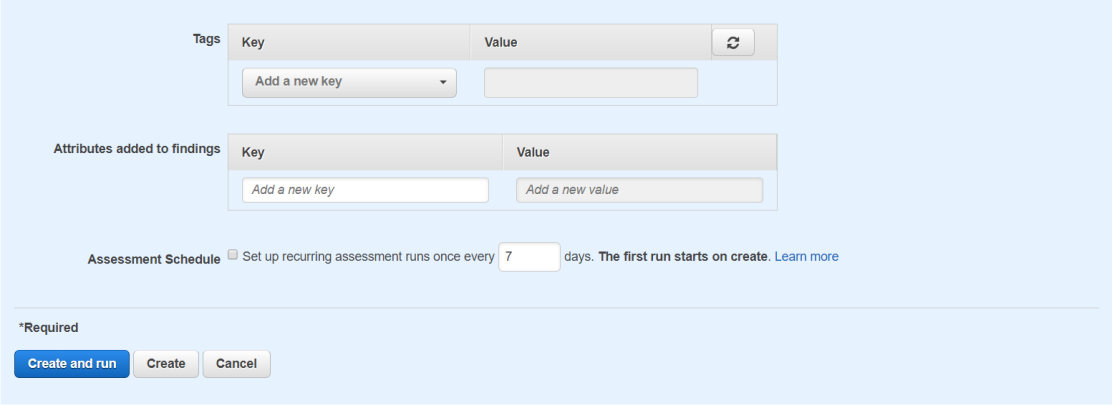


Figure 18. Finalizing an assessment template

5.2 Analyzing assessment findings

After an assessment has run successfully, its results can be accessed from Inspector Console view. Assessment runs-tab lists all completed assessments with different characteristics of them as columns. Figure 19 illustrates how users can either expand a particular assessment to see further details or a report can be downloaded. Users can choose between downloading a full report or a findings report.

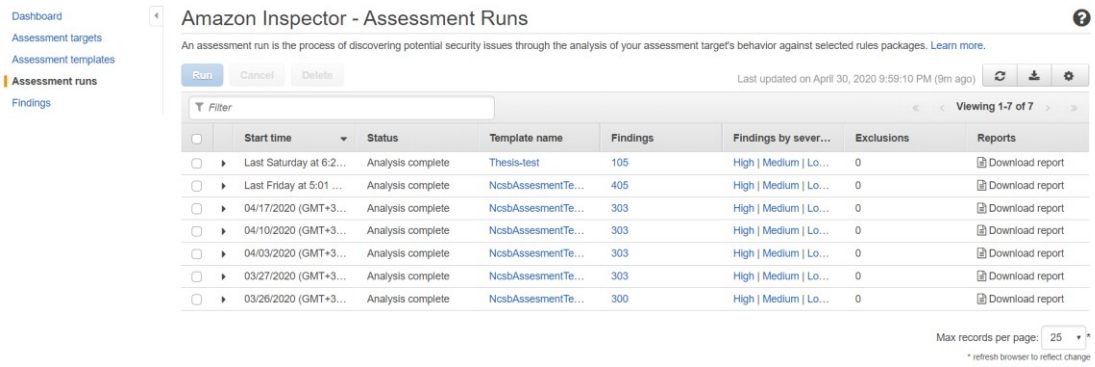
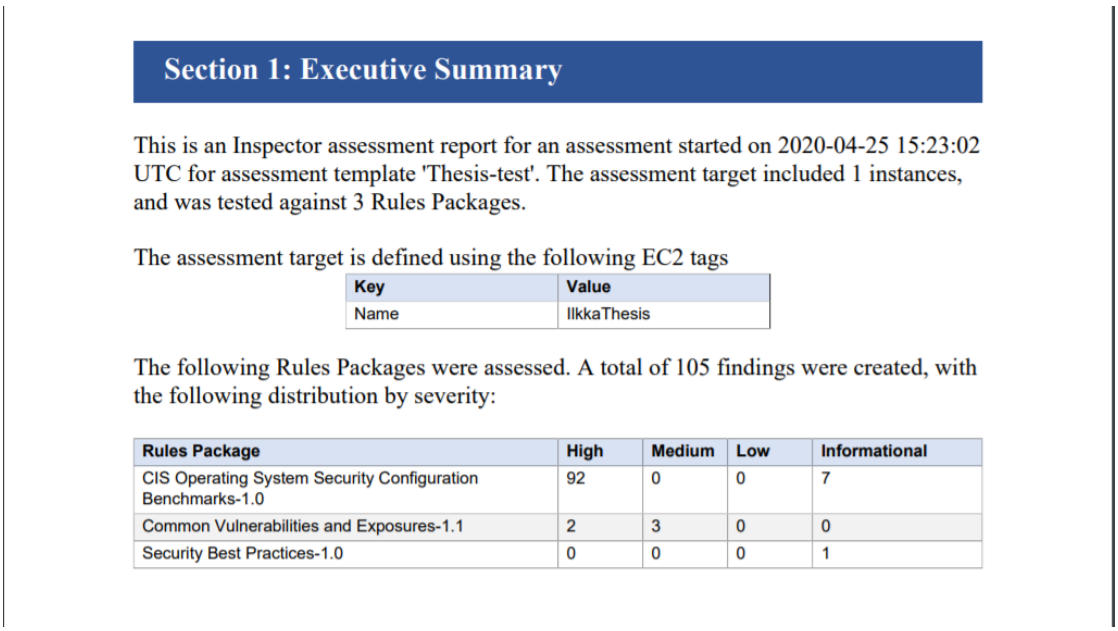


Figure 19. Accessing assessment results

The full report is as its name suggests extremely comprehensive. The test assignment generated 105 findings and a 1021-page full report was generated from them.

Both reports include a summary, list of EC2 instances that were assessed, names of rule packages against which instances were evaluated against and information on findings alongside with respective mitigation steps. The full report has additionally each rule in rules packages listed.

Figure 20 shows an executive summary which can be found at the beginning of each report. It has general information of the assessment and findings count.



Section 1: Executive Summary

This is an Inspector assessment report for an assessment started on 2020-04-25 15:23:02 UTC for assessment template 'Thesis-test'. The assessment target included 1 instances, and was tested against 3 Rules Packages.

The assessment target is defined using the following EC2 tags

Key	Value
Name	IlkkaThesis

The following Rules Packages were assessed. A total of 105 findings were created, with the following distribution by severity:

Rules Package	High	Medium	Low	Informational
CIS Operating System Security Configuration Benchmarks-1.0	92	0	0	7
Common Vulnerabilities and Exposures-1.1	2	3	0	0
Security Best Practices-1.0	0	0	0	1

Figure 20. Executive summary of an assessment

Section two lists rules packages and assessed instances while section three has findings in a table format. The table columns consist of rules that generated the finding, the severity of the finding and how many instances were affected. Since in the example only one instance was tested Figure 21 has a constant value of one in *failed* column.

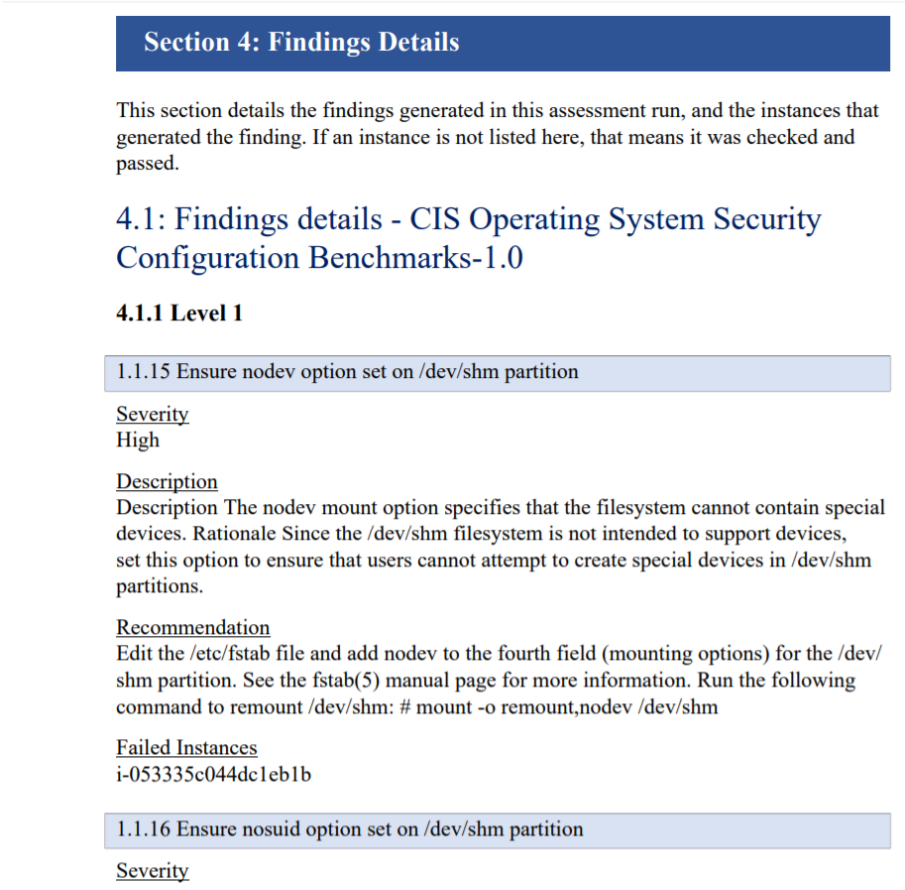
3.1: Findings table - CIS Operating System Security Configuration Benchmarks-1.0

3.1.1 Level 1

Rule	Severity	Failed
1.1.15 Ensure nodev option set on /dev/shm partition	High	1
1.1.16 Ensure nosuid option set on /dev/shm partition	High	1
1.1.17 Ensure noexec option set on /dev/shm partition	High	1
1.1.1.1 Ensure mounting of cramfs filesystems is disabled	High	1
1.1.1.2 Ensure mounting of freevxfs filesystems is disabled	High	1
1.1.1.3 Ensure mounting of jffs2 filesystems is disabled	High	1
1.1.1.4 Ensure mounting of hfs filesystems is disabled	High	1
1.1.1.5 Ensure mounting of hfsplus filesystems is disabled	High	1
1.1.1.6 Ensure mounting of squashfs filesystems is disabled	High	1
1.1.1.7 Ensure mounting of udf filesystems is disabled	High	1
1.1.1.8 Ensure mounting of FAT filesystems is disabled	High	1
1.2.3 Ensure gpgcheck is globally activated	High	1
1.3.1 Ensure AIDE is installed	High	1
1.3.2 Ensure filesystem integrity is regularly checked	High	1
1.4.1 Ensure permissions on bootloader config are configured	High	1
1.4.2 Ensure authentication required for single user mode	High	1

Figure 21. Findings table

Section four has more detailed information on findings and how to mitigate found security issues. Each single finding follows the format depicted in Figure 22. First the security check or rule is announced alongside with its severity. *Description* gives more information on found issue and *Recommendation* lays out mitigation steps. Finally, all instances that failed the rule in an assessment are listed.



Section 4: Findings Details

This section details the findings generated in this assessment run, and the instances that generated the finding. If an instance is not listed here, that means it was checked and passed.

4.1: Findings details - CIS Operating System Security Configuration Benchmarks-1.0

4.1.1 Level 1

1.1.15 Ensure nodev option set on /dev/shm partition

Severity
High

Description
Description The nodev mount option specifies that the filesystem cannot contain special devices. Rationale Since the /dev/shm filesystem is not intended to support devices, set this option to ensure that users cannot attempt to create special devices in /dev/shm partitions.

Recommendation
Edit the /etc/fstab file and add nodev to the fourth field (mounting options) for the /dev/shm partition. See the fstab(5) manual page for more information. Run the following command to remount /dev/shm: # mount -o remount,nodev /dev/shm

Failed Instances
i-053335c044dc1eb1b

1.1.16 Ensure nosuid option set on /dev/shm partition

Severity

Figure 22. Security finding format

As Figure 19 showed, a grand total of 105 findings or security issues with varying severity were found in the assessment. Going through all of them in great depth alongside with mitigation steps could be a thesis subject of its own and well beyond the scope of this thesis. However, to illustrate how practical and easy it is to follow the given recommendation steps in a report a few findings will be shown as an example.

Ensure default deny firewall policy

Denying connections by default on the firewall is predicated on the basis that it is easier to allow or whitelist legitimate traffic than it is to deny or blacklist illegitimate traffic. This reasoning is purely a mathematical one: the number of potential attack vectors versus the number of common legitimate use cases in the environment heav-

ily favors the former. This setting also minimizes the risks caused by accidental firewall misconfigurations. AWS' recommended remediation is to run the following commands to set iptables' policies to drop everything be default:

- *sudo iptables -P INPUT DROP*
- *sudo iptables -P OUTPUT DROP*
- *sudo iptables -P FORWARD DROP*

Ensure SSH root login is disabled

Users and administrators are forced to login using their own accounts and then escalating privileges to root using *sudo* or *su* should they need to. This leaves an audit trail behind which for example in case of a security breach is extremely valuable information. The trail can be used to track down which user was directly responsible or had their account compromised, which was then used to gain access to the system. Figure 23 exhibits the configuration change needed to be done in */etc/ssh/sshd_config*-file to achieve this.

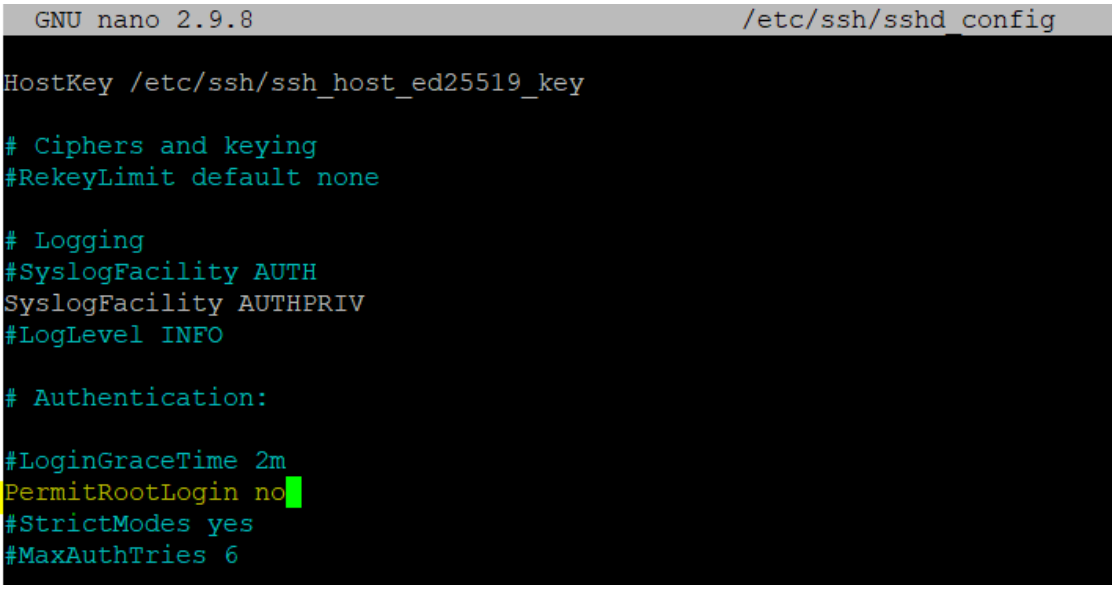


Figure 23. Disabling SSH root-login

Ensure password creation requirements are conFigured

Enforcing strong password policies is a key attribute to not having users' credentials susceptible to brute forcing. *Pam_pwquality.so*-module can and should be used to assure this. The recommended steps include configuring the module in two files. The first file */etc/pam.d/password-auth* has correct configuration by default which is having "pam_pwquality.so", "try_first_pass" and "retry=3" declared in password requisite-row. */etc/security/pwquality.conf*-file configuration changes are depicted in Figure 24 which forces passwords to have both lower- and uppercase letters, numbers, and a special character in them with the minimum length of 14. These values are not set in stone; they can be customized arbitrarily just as long they *are* in use.

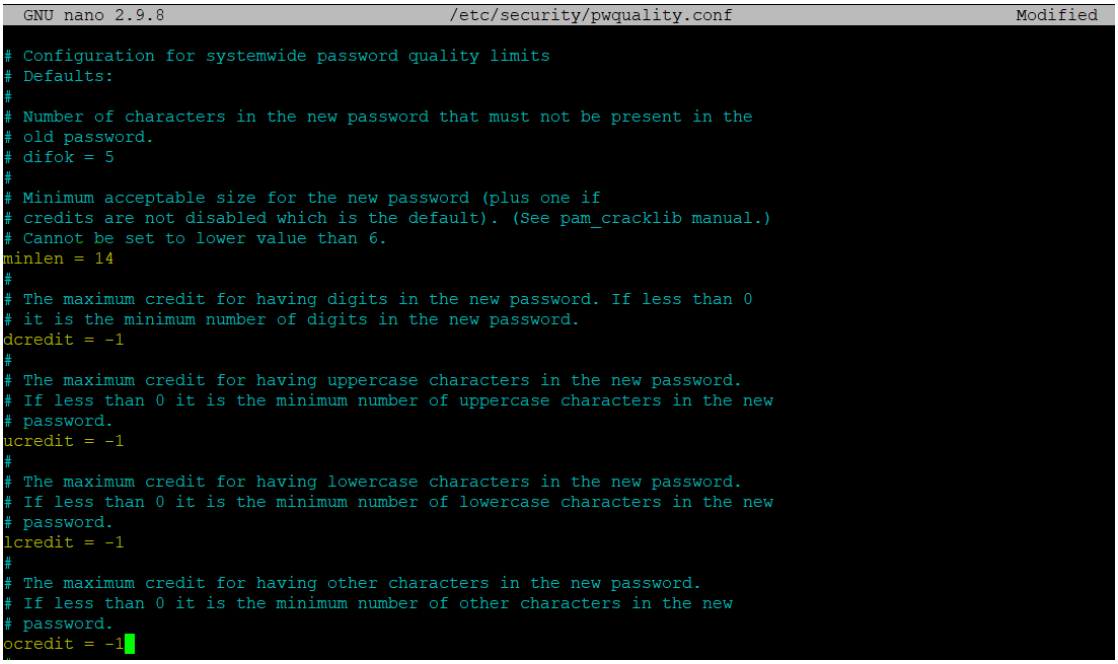


Figure 24. Strong password enforcement

Ensure default user umask is 027 or more restrictive

When a file is created on the system it is created with some default permissions. Umask is used to control those default permissions. Hardening them has the benefit of files and directories not being accessible by others by accident. To make directories and files accessible to others, users must deliberately set permissions accord-

ingly. Umask can be configured in standard shell configuration files. Umask permissions are stated in reverse from normal permissions, i.e. if a file has permissions of 664, using umask permission declaration the permissions are 113 (777-664 = 113). Recommendation states that default user umask should be 027 or more restrictive which means that when a file or directory is created, it is created with permissions of 750 or less. To implement this configuration, "*umask 027"* is added or edited in the shell's configuration file that is in use, e.g. if bash is used, */etc/bashrc* is edited.

Ensure access to the su command is restricted

Users can use *su*-command to run commands or shells as another user. It can also be used to escalate privileges to root-user temporarily. Therefore, it is paramount that when it is used an audit trail would be left behind of its use. Unfortunately, this is not the case with *su* as when it is run, a trail of it only reveals that a user executed it. *su* has been replaced by *sudo* which offers much better auditing possibilities. Hence, restricting access to *su*-command is recommended. To start, */etc/pam.d/su*-file contains a default comment "auth required pam_wheel.so use_uid" which should be uncommented. This results in users who are in *wheel*-group only being able to run *su*-command.

6 GuardDuty use case: catching potentially malicious activity

In a situation where an account or a resource has been compromised without any indication or knowledge of the compromise, the damages can quickly result in major economic losses or worse yet, in complete interruption of business continuity or personal data losses. This has been the case where adversaries have first managed to compromise a single host in a corporate network and from there through lateral movement and privilege escalation ultimately have crippled the whole network, e.g. with ransomware. Having automated processes and services scanning for anomalies both in network and in resource usage, can go a long way in protecting businesses against these threats.

GuardDuty provides such service. As an example use case, let it be assumed that a user has unknowingly misplaced or lost a USB-stick containing private SSH-key and information regarding EC2-instance which is used for work-related tasks. Next, this stick ends up in the hands of a malicious actor who cannot resist the temptation of trying to exploit the situation, having direct access to a corporate asset. To illustrate the situation, an EC2-instance was setup with SSH enabled.

When an adversary manages to get illegitimate access to an asset, the first points of interest for the intruder often include what user he is logged in as, what privileges that user has, what other users might be present on the system and what their privileges are. The process of gathering this information is called *user enumeration*. The goal is to move laterally in the environment to discover vulnerabilities which might ultimately lead into privilege escalation, which means gaining more privileges than the original compromised user had.

Using AWS *Command Line Interface* or CLI, users have the option to manage their services and resources, e.g. EC2-instances. Users can also use it manage and enumerate users by making API-calls via AWS CLI. The following list contains just some of the commands which could be used to collect sensitive information which adversaries could use for both lateral movement and privilege escalation. (AWS IAM API calls 2020)

- *aws iam get-user*
- *aws iam get-user-policy*
- *aws iam list-users*
- *aws iam list-groups*
- *aws iam list-roles*
- *aws iam list-user-policies*

Let it also be assumed that the user whose private key got compromised does not do any administrative work and hence has no reason to be making these kind of API calls. Suddenly, these API calls are originating from user's EC2-instance. It is these kinds of anomalies that GuardDuty is constantly trying to detect based on past observed behavior. For testing purposes, these commands were run on the test EC2-instance without running them on it earlier to simulate the use case.

As can be seen in Figure 25, GuardDuty has considered this behavior from this entity to be anomalous and has raised a finding or an alarm about it.

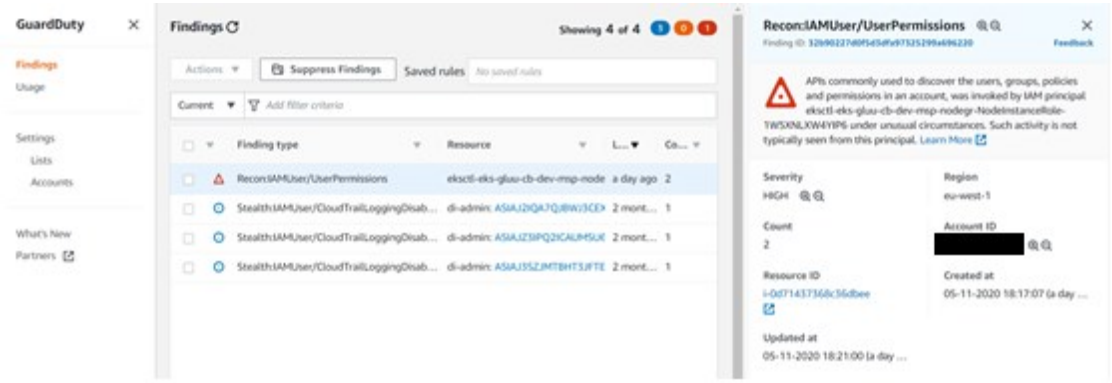


Figure 25. GuardDuty finding of anomalous activity

What is of note is the description of the event which explains why GuardDuty picked up on this activity.

> *APIs commonly used to discover the users, groups, policies and permissions in an account, was invoked by IAM principal <principal_name_concealed> under unusual circumstances. Such activity is not typically seen from this principal.*

The finding has more relevant information regarding the incident available such as the exact timestamp of when the activity took place, source IP, network information, instance ID and what API calls were made, to name a few. This information can be used by administrators to quickly assess the situation. In this case, it would help them come to the realization that they have a compromised host in their environment and thus, make correct decisions on how to proceed in taking mitigating actions. These actions could include isolating the host from the network entirely and taking snapshots and images of the compromised host for forensic analysis to determine how the intruder got in and what impact the breach might have on business and related processes. Comparing this situation with the situation where automatic detection of a breach is not available it becomes trivial to see the benefits GuardDuty and other anomaly-based detection security services can offer.

In chapter 3.5 it was discussed how Security Hub can be configured to forward security findings in various ways. The use case illustrated here provides a perfect example how this could be utilized. Figure 25 showcased that the severity of the GuardDuty finding was “high”. Since GuardDuty findings are sent to Security Hub, Security Hub could be configured to send an email or SMS-message to administrators once it receives a high-severity finding from GuardDuty. In the use case, this would mean that once the adversary started enumerating users on the compromised host, administrators would have received a notification about it at the same time, giving them infinitely better chances to take control of the situation and minimize potential damages.

7 Conclusions

With over ten security-oriented services alongside with virtual private cloud possibilities as well as identity and access management, not to mention secure resource management, AWS security can get extremely multifaceted. New users especially might feel overwhelmed with all the possible choices to choose from. To help with all this, AWS has introduced centralized control mechanisms or services to facilitate the process of governing users' varying environments and securing them. *Trusted Advisor* was introduced as a tool to help users manage their accounts and services and guide them in resource management and optimization. It consists of five different categories with emphasis in environment optimization. One of the categories is centered around security and it provides universal guidelines to various security areas. However, Trusted Advisor is not a security service per se, and especially its free tier is inadequate in securing vast multilayered AWS cloud solutions.

EC2 or Elastic Compute Cloud is a flagship product of AWS and it is only fitting a specially designed security service *Inspector* was developed to assess the security of its instances. It is extremely easy to use and automating it to assess large fleet of instances has been made effortless. Currently, it comes with four different rules packages which are comprehensive to say the least. Unlike in the example assessment configured and ran in chapter 5, the full potential of Inspector comes to fruition when the whole process of mitigation of security findings is automated through CloudWatch and Lambda. Large cloud infrastructures can have thousands and thousands of instances running simultaneously. The drudgery of having to go through each and every one of them and implement fixes laid out in assessments' generated reports is not a feasible one. Fortunately, this can be avoided. Inspector can also be considered a great tool for learning about security and hardening one's servers and applications. Reports that are generated by assessment runs are wonderfully structured and easy to follow even for users that are still new to security and hardening processes of servers and applications.

GuardDuty can be thought of as AWS' SIEM-system in the cloud. It collects log data from CloudTrail, VPCs or virtual private clouds and DNS. After, findings are produced

when received logs match a predetermined set of criteria. This is a two-fold proposition. On one hand, it takes the burden from users of having to think and maintain rules which can get quite complex. On the other hand, this leaves existing rules somewhat obscure since there is no visibility into how rules are constructed and what is the underlying logic behind them. Perhaps the biggest shortcoming of GuardDuty is the impossibility of generating own findings or even finding types. Even though the platform and used services might be the same from one environment to another, no two environments are *exactly* the same. This is especially true in corporate environments. Each company which decides to host their services in AWS is a unique entity with their own unique requirements in security monitoring due to different roles, users, used services, threat landscape, potential attack vectors, internal threats and configurations, to name a few. All these variables demand customizability in security solutions and unfortunately, GuardDuty comes up short.

Security Hub offers a centralized management portal to both internal AWS security services and external third-party security solutions. By collecting security events and findings from multiple sources, Security Hub offers a wider general view into users' environments security posture. Findings discovered in the assessment in chapter 5.1 could also be retrieved from Security Hub as Figure 26 illustrates.

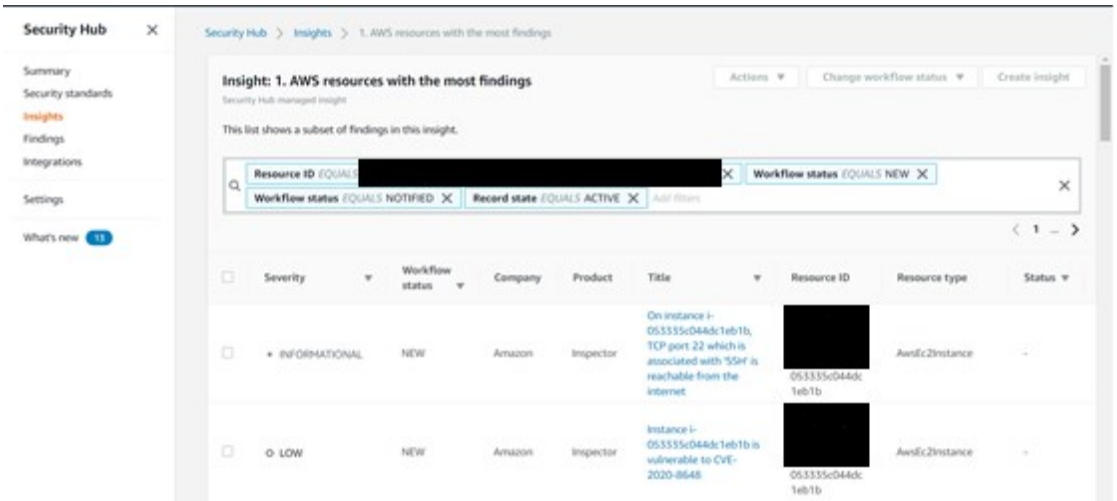


Figure 26. Inspector assessment findings in Security Hub

Security Hub does not require vast expertise to use at all. It has been made easily approachable to any user and is intuitive to use. Amidst of a major security incident Security Hub can offer a quick overall view where major problem areas might be and where to invest resources to try and get a handle on the situation.

On 31st of March 2020, AWS announced a new security tool for investigating security issues called *AWS Detective* being publicly accessible. (Detective 2020) Detective shares many of the attributes with services researched in this thesis. It collects log data and analyzes them using machine learning and statistical analysis. As a new feature it also takes in CloudTrail and GuardDuty findings on top of log-data. Detective aims at helping users to conduct investigations into security issues and determine root causes behind them. To achieve this, it extracts users and resources from log-data and findings produced by another services and forms relationships between them. During a security incident this enables users to quickly make educated assumptions into what might be the root cause of the incident which in turn enables more efficient resource allocation into the incident and faster remediation times.

Deciding what strategy to take and what services to choose from a wide range of available security services is by no means an easy task. The setup researched here is just one implementation in how to monitor security in AWS public cloud. It offers basic widescale coverage of various security topics. Trusted Advisor and GuardDuty cover account and service management while Inspector is used to monitor and assess EC2-instances. GuardDuty serves additional coverage as well. Security Hub gathers findings from Inspector and GuardDuty (with other potential integrated services as well) to provide a centralized, situational perspective into the overall security status quo. As stated, it covers basic areas; to have a more holistic coverage of the environment other security controls should not be forgotten, such as CloudTrail or Macie.

The goal of this thesis was to research and document the preselected set of security monitoring tools. During the process it became abundantly clear the researched setup was very easy to use. It does not require users to be AWS power users to being

able to take advantage of all the guidance and suggestions it offers. More importantly, when potential issues are discovered by it, easy-to-follow and instructive remediation steps are provided. While the lab environment used in this thesis had the tools preinstalled, installing them oneself is not difficult at all. Installation can be done from their respective console views and with just a few clicks on a mouse, AWS takes care of all the rest. The practical part of this thesis, namely chapters 4 to 6, should offer new users clear steps how these tools can be utilized. More advanced users most likely are already familiar with them. Next area for research could include how to automate all the processes described here. Especially automating mitigation processes would offer a challenging but interesting field of study. There are endless ways of how to implement this automation, user's imagination being the biggest limiter. Therefore, research in this field could provide even advanced users valuable insights how to implement automation in their environments.

References

AWS CloudWatch. 2020. Amazon documentation. Accessed on 12 February 2020. Retrieved from https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html

AWS IAM API calls. 2020. Amazon documentation. Accessed on 13 May 2020. Retrieved from https://docs.aws.amazon.com/cli/latest/reference/iam/

AWS KMS. 2020. AWS Key Management Service. Accessed on 10 February 2020. Retrieved from https://aws.amazon.com/kms/

AWS Shield. 2020. Amazon documentation. Accessed on 12 February 2020. Retrieved from https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html

AWS VPC. 2020. AWS Virtual Private Cloud. Amazon documentation. Accessed on 10 February 2020. Retrieved from https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html

AWS WAF. 2020. AWS Web Application Firewall. Amazon documentation. Accessed on 11 February 2020. Retrieved from https://docs.aws.amazon.com/waf/latest/developerguide/what-is-aws-waf.html

Best practice checklist. 2020. Amazon web page. Accessed on 23 February 2020. Retrieved from https://aws.amazon.com/premiumsupport/technology/trusted-advisor/best-practice-checklist/

Carey, S. 2019. The history of AWS. Accessed on 4 January 2020. Retrieved from https://www.computerworld.com/article/3412382/the-history-of-aws--a-timeline-of-defining-moments-from-2002-to-now.html#slide2

Cloud computing benefits. 2020. Cloud computing beginner's guide on Microsoft's web page. Accessed on 10 January 2020. Retrieved from https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/

CloudWatch. 2020. Amazon documentation. Accessed on 12 February 2020. Retrieved from https://aws.amazon.com/blogs/startups/monitoring-an-app-examples-from-the-aws-startup-kit/

CloudWatch events. 2020. Amazon documentation. Accessed on 13 May 2020. Retrieved from https://docs.aws.amazon.com/awssupport/latest/user/cloudwatch-events-ta.html

Davis, J. 2019. Ransomware Attacks Double in 2019. Accessed on 10 January 2020. Retrieved from https://healthitsecurity.com/news/ransomware-attacks-double-in-2019-brute-force-attempts-increase

Detective. 2020. Amazon Press release. Accessed on 3 May 2020. Retrieved from https://press.aboutamazon.com/news-releases/news-release-details/aws-announces-general-availability-amazon-detective

ESDS. 2018. Cloud types. Accessed on 9 January 2020. Retrieved from https://www.esds.co.in/blog/cloud-computing-types-cloud/

Fitzgerald, E. 2018. How to remediate Amazon Inspector Security Findings Automatically. Blogpost on Amazon's website. Accessed on 21 February 2020. Retrieved from https://aws.amazon.com/blogs/security/how-to-remediate-amazon-inspector-security-findings-automatically/

Griffith, E. 2016. What Is Cloud Computing? Accessed on 9 January 2020. Retrieved from https://uk.pcmag.com/networking-communications-software/16824/what-is-cloud-computing

GuardDuty. 2020. Amazon documentation. Accessed on 21 February 2020. Retrieved from https://docs.aws.amazon.com/guardduty/latest/ug/guardduty-ug.pdf

GuardDuty findings. 2020. Amazon documentation. Accessed on 17 April 2020. Retrieved from https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_finding-types-active.html

History of AWS. N.d. Accessed on 4 January 2020. Retrieved from https://www.javatpoint.com/history-of-aws

Inspector. 2020. Amazon documentation. Accessed on 21 February 2020. Retrieved from https://docs.aws.amazon.com/inspector/latest/userguide/inspector_introduction.html

Inspector agents. 2020. Amazon documentation. Accessed on 24 April 2020. Retrieved from https://docs.aws.amazon.com/inspector/latest/userguide/inspector_installing-uninstalling-agents.html

Macie. 2020. Amazon's web page. Accessed on 12 February 2020. Retrieved from https://aws.amazon.com/macie/

Rojas, A. 2018. A Brief History of AWS. Referenced on 4 January 2020. Retrieved from https://mediatemple.net/blog/news/brief-history-aws/

Security Hub. 2020. Amazon documentation. Accessed on 23 February 2020. Retrieved from https://docs.aws.amazon.com/securityhub/latest/userguide/securityhub-concepts.html

Security Hub Standards. 2020. Amazon documentation. Accessed on 17 April 2020. Retrieved from https://docs.aws.amazon.com/securityhub/latest/userguide/securityhub-standards.html

Stalcup, K. 2020. AWS vs Azure vs Google Cloud Market Share 2020: What the Latest Data Shows. Blogpost on ParkMyCloud's web page. Published on 5 February 2020. Accessed on 7 May 2020. Retrieved from https://www.parkmycloud.com/blog/aws-vs-azure-vs-google-cloud-market-share/

Trusted Advisor. 2020. Amazon web page. Accessed on 23 February 2020. Retrieved from https://aws.amazon.com/premiumsupport/technology/trusted-advisor/

VPC. 2020. Amazon documentation. Accessed on 10 February 2020. Retrieved from https://docs.aws.amazon.com/vpc/latest/userguide/vpce-gateway.html

Yadav, K. 2018. AWS services. Accessed on 6 February 2020. Retrieved from https://blog.usejournal.com/what-is-aws-and-what-can-you-do-with-it-395b585b03c

Weiss, B. 2019. AWS re:Inforce 2019: The fundamentals of AWS Cloud Security. AWS Conference talk. Accessed on 9 February 2020. Retrieved from https://www.youtube.com/watch?v=-ObImxw1PmI

What is IAM? 2020. Amazon documentation. Accessed on 10 February 2020. Retrieved from https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html